БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
Ханты-Мансийского автономного округа – Югры
«Сургутский государственный университет»

Политехнический институт
«Цифровая кафедра»

УТВЕРЖДАЮ
Первый проректор
_____ Даниленко И.Н.
«15» июня 2022 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ
(далее - ДПП ПП, образовательная программа, программа)

«Программирование и базы данных»

Трудоемкость: 648 академических часов (18 зет.)

Сургут – 2022

Авторы программы:
Кузин Дмитрий Александрович, к.т.н., доцент _____

Программа рассмотрена и одобрена на заседании учебно-методического совета Политехнического института «10» 06 2022 года, протокол № 05/22
Председатель УМС ПИ _____ Е.Н. Паук

1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Цель освоения программы

Подготовка специалистов, получающих новую квалификацию в области информационных технологий.

1.2. Характеристика новой квалификации и связанных с ней видов профессиональной деятельности, трудовых функций и уровней квалификации

Программа разработана в соответствии с профессиональным стандартом Программист (Приказ Министерства труда и социальной защиты Российской Федерации от 18 ноября 2013 г.; № 679н); Концепцией реализации результата «Обучающимся обеспечена возможность прохождения профессиональной переподготовки в рамках проекта «Цифровые кафедры» образовательной организации высшего образования – участника программы стратегического академического лидерства «Приоритет-2030» посредством получения дополнительной квалификации по ИТ-профилю» федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика Российской Федерации»

1.2.1. Наименование вида профессиональной деятельности

Разработка программного обеспечения.

1.2.2. Основная цель вида профессиональной деятельности

Разработка, отладка, проверка работоспособности, модификация программного обеспечения.

1.2.3. Описание трудовых функций, входящих в профессиональный стандарт (функциональная карта вида профессиональной деятельности)

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
A	Разработка и отладка программного кода	3	Формализация и алгоритмизация поставленных задач	A/01.3	3
			Написание программного кода с использованием языков программирования, определения и манипулирования данными	A/02.3	3
			Оформление программного кода в	A/03.3	3

			соответствии с установленными требованиями		
			Работа с системой контроля версий	A/04.3	3
			Проверка и отладка программного кода	A/05.3	3

1.3. Категория слушателей: к освоению ДПП ПП допускаются лица, имеющие или получающие высшее профессиональное образование, освоившие основную профессиональную образовательную программу (далее – ОПОП ВО) бакалавриата – в объеме не менее первого курса (бакалавры 2-го курса), ОПОП ВО специалитета – не менее первого и второго курсов (специалисты 3-го курса), обучающиеся по ОПОП ВО, не отнесенным к ИТ-сфере.

1.3. Трудоемкость: 648 часов (18 зет).

1.4. Сроки освоения программы:
теоретическое обучение: 75 недель.

1.5. Форма реализации программ: очно-заочная. При реализации программы применяются дистанционные образовательные технологии и электронное обучение.

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

2.1. Перечень новых компетенций, формирующихся в результате освоения программы

ПК-1. Применяет языки программирования для решения профессиональных задач;
ПК-2. Применяет принципы и основы алгоритмизации;
ПК-3. Применяет системы управления базами данных (СУБД).

2.2. Перечень знаний, умений и навыков

По окончании обучения слушатели должны **знать:**
- Методы и приемы формализации задач;
- Языки формализации функциональных спецификаций;
- Методы и приемы алгоритмизации поставленных задач;
- Нотации и программные продукты для графического отображения алгоритмов;
- Алгоритмы решения типовых задач, области и способы их применения;
- Синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования;
- Методологии разработки программного обеспечения;
- Методологии и технологии проектирования и использования баз данных;
- Технологии программирования;
- Особенности выбранной среды программирования и системы управления базами данных;
- Компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними.

По окончании обучения слушатели должны **уметь:**

- Использовать методы и приемы формализации задач;
- Использовать методы и приемы алгоритмизации поставленных задач;
- Использовать программные продукты для графического отображения алгоритмов;
- Применять стандартные алгоритмы в соответствующих областях;
- Применять выбранные языки программирования для написания программного кода;
- Использовать выбранную среду программирования и средства системы управления базами данных;
- Использовать возможности имеющейся технической и/или программной архитектур

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Мес	Сентябрь					Октябрь				Ноябрь				Декабрь					Январь				Февраль				Март					Апрель				Май					Июнь					Июль				Август			
Числа	1 - 7	8 - 14	15 - 21	22 - 28	29 - 5	6 - 12	13 - 19	20 - 26	27 - 2	3 - 9	10 - 16	17 - 23	24 - 30	1 - 7	8 - 14	15 - 21	22 - 28	29 - 4	5 - 11	12 - 18	19 - 25	26 - 1	2 - 8	9 - 15	16 - 22	23 - 1	2 - 8	9 - 15	16 - 22	23 - 29	30 - 5	6 - 12	13 - 19	20 - 26	27 - 3	4 - 10	11 - 17	18 - 24	25 - 31	1 - 7	8 - 14	15 - 21	22 - 28	29 - 5	6 - 12	13 - 19	20 - 26	27 - 2	3 - 9	10 - 16	17 - 23	24 - 31
Нед	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
I	=																	*	*	Э	Э																Э	Э	К	К	К	К	К	К	К	К	К	К	К	К
II	=																	*	*	Э	Э	П	П	Г	Г	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=

Сводные данные

		Курс 1			Курс 2			Итого
		Сем. 1	Сем. 2	Всего	Сем. 3	Сем. 4	Всего	
	Теоретическое обучение	16	16	32	16		16	48
Э	Экзаменационные сессии	2	2	4		2	2	6
П	Практика					2	2	2
Г	Итоговая аттестация					2	2	2
К	Каникулы		13	13				13
*	Нерабочие праздничные дни (не включая воскресенья)	2(12 дн)		2(12 дн)	2(12 дн)		2(12 дн)	4(24 дн)
Продолжительность обучения (не включая нерабочие праздничные дни и каникулы)		не менее 12 нед. ине более 39 нед.			не менее 12 нед. ине более 39 нед.			
Итого		20	31	**51**	18	6	**24**	75
Студентов								
Групп								

Курс	Сентябрь					Октябрь					Ноябрь				Декабрь					Январь				Февраль				Март					Апрель				Май					Июнь					Июль				Август			
		5	12	19	26	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	6	13	20	27	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28
		6	13	20	27	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	31	7	14	21	28	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29
		7	14	21	28	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	1	8	15	22	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30
	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	2	9	16	23	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	31
	2	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	3	10	17	24	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	
	3	10	17	24	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	
	4	11	18	25	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	5	12	19	26	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
1																				Э	Э	Э																Э	Э	Э	Э	К	К	К	К	К	К	К	К	К	К	К	К
																			Э																																			
2																				Э	Э	Э	П	П	Г	Г	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=
																			Э																																			
																			Э																																			
																			Э																																			

4. УЧЕБНЫЙ ПЛАН

	Наименование	Форма контроля		з.е.		Часов в з.е.	Итого акад.часов				Курс 1								Курс 2				Компетенции
											Семестр 1				Семестр 2				Семестр 3				
		Экзамен	Зачет	Экспертное	Факт		Экспертное	По плану	Конт. раб.	СР	з.е.	Лек	Пр	СР	з.е.	Лек	Пр	СР	з.е.	Лек	Пр	СР	
Дисциплины (модули)				**18**	**18**		**648**	**648**	**96**	**552**	**4**		**32**	**112**	**4**		**32**	**112**	**10**		**32**	**328**	
01	Программирование на языке Python		1	4	4	36	144	144	32	112	4		32	112									ПК-1; ПК-2
02	Работа с данными на языке Python		2	4	4	36	144	144	32	112					4		32	112					ПК-3; ПК-2
03	Разработка веб-приложений на языке Python		3	4	4	36	144	144	32	112									4		32	112	ПК-3; ПК-1
04	Практика		3	3	3	36	108	108		108									3			108	ПК-3; ПК-1; ПК-2
05	Итоговая аттестация			3	3	36	108	108		108									3			108	ПК-3; ПК-1; ПК-2

5. РАБОЧИЕ ПРОГРАММЫ УЧЕБНЫХ ДИСЦИПЛИН (МОДУЛЕЙ) ПРАКТИК

5.1 Рабочая программа дисциплины «Программирование на языке Python»

5.1.1 Структура дисциплины

№ п/п	Наименование элементов программы	Всего часов	Практические/ лабораторные работы	Сам. раб.	Формируемые компетенции
1.	**Раздел 1. Основы программирования на языке Python**				
1.1.	Установка Python. Знакомство с особенностями сред программирования на Python.	10	2	8	ПК-2
1.2	Введение в методологии разработки программного обеспечения. Ввод-вывод и переменные в Python.	8	2	6	ПК-1, ПК-2
1.3.	Операции и выражения. Условный оператор. Основы работы со строками	8	2	6	ПК-1, ПК-2
1.4.	Операторы цикла for, while	6	2	4	ПК-1, ПК-2
2.	**Раздел 2. Функции и модули**				
2.1.	Функции. Области видимости переменных. Параметры	12	2	10	ПК-1, ПК-2
2.2.	Функции с переменным числом аргументов	8	2	6	ПК-1, ПК-2
2.3.	Функции как объект. Лямбда-функции. Декораторы.	8	2	6	ПК-1, ПК-2
2.4.	Модули стандартной библиотеки языка Python.	12	2	10	ПК-1, ПК-2
3.	**Раздел 3. Работа с коллекциями**				
3.1.	Множества. Кортежи	12	2	10	ПК-1, ПК-2
3.2.	Строки. Списки. Срезы	12	2	10	ПК-1, ПК-2
3.3.	Методы split и join. Списочные выражения	6	2	4	ПК-1, ПК-2

3.4.	Словари. Функции map и filter	10	2	8	ПК-1, ПК-2
4.	**Раздел 4. Объектно-ориентированное программирование**				
4.1.	Введение в технологии программирования. ООП. Полиморфизм	12	2	10	ПК-1, ПК-2
4.2.	ООП. Определение операторов	6	2	4	ПК-1, ПК-2
4.3.	ООП. Наследование	6	2	4	ПК-1, ПК-2
4.4.	Проектирование и разработка классов	8	2	6	ПК-1, ПК-2
	ИТОГО:	**144**	**32**	**112**	ПК-1, ПК-2

5.1.2 Содержание дисциплины

Раздел 1. Основы программирования на языке Python.

Тема 1.1. Установка Python. Знакомство с особенностями сред программирования на Python.

1. Основные этапы компьютерного решения задачи.
2. Краткий обзор языков формальной спецификации.
3. Функциональные диаграммы, диаграммы переходов состояний.
4. Понятие алгоритма и его свойства.
5. Способы записи алгоритмов: формульное описание, блок-схема, словесное описание в виде набора правил.
6. Блок-схема: основные элементы и правила построения.
7. Разновидности алгоритмов: линейные, разветвляющиеся, циклические.
8. Обзор редакторов кода и сред разработки для Python.
9. Особенности установки и работы в стандартной среде Python.
10. Особенности установки и работы в IDE JetBrains PyCharm.

Тема 1.2. Введение в методологии разработки программного обеспечения. Ввод-вывод и переменные в Python.

1. Жизненный цикл программного продукта.
2. Обзор характерных особенностей основных методологий разработки программного обеспечения: каскадная модель, V-модель, инкрементная модель, модель быстрой разработки программного обеспечения (RAD-модель), гибкая (Agile) методология, итеративная модель, спиральная модель.
3. Методики разработки алгоритмов.
4. Типы данных и переменные в Python.
5. Функции ввода и вывода в Python.

Тема 1.3. Операции и выражения. Условный оператор. Основы работы со строками.

1. Обзор возможностей программ и бесплатных онлайн-сервисов для создания блок-схем алгоритмов.
2. Построение графического представления алгоритма в онлайн-сервисе draw.io.
3. Арифметические операции и выражения.
4. Основные операции над строками.
5. Логические операции и выражения.

6. Условный оператор, вложенный условный оператор, конструкция «иначе-если».

Тема 1.4. Операторы цикла for, while.
1. Цикл for
2. Цикл while
3. Команды управления циклом.

Раздел 2. Функции и модули
Тема 2.1. Функции. Области видимости переменных. Параметры.
1. Понятие и назначение функции в программировании.
2. Встроенные функции в Python.
3. Примеры параметров встроенных функций.
4. Синтаксис создания пользовательской функции.
5. Порядок передачи аргументов в функцию.
6. Особенности вызова функции.
7. Возврат функцией нескольких значений.
8. Понятие и особенности работы с областью видимости.

Тема 2.2. Функции с переменным числом аргументов.
1. Встроенные функции с переменным числом аргументов.
2. Определение пользовательской функции с переменным числом позиционных аргументов.
3. Организация доступа к значениям аргументов в функции с переменным числом позиционных аргументов.
4. Определение пользовательской функции с переменным числом именованных аргументов.
5. Организация доступа к значениям аргументов в функции с переменным числом именованных аргументов.

Тема 2.3. Функции как объект. Лямбда-функции. Декораторы.
1. Особенности работы с функциями как с объектами в языке Python.
2. Понятие анонимной функции (лямбда-функции).
3. Лямбда-функция: назначение, синтаксис, примеры использования.
4. Декораторы: определение и назначение.
5. Синтаксис объявления декоратора.
6. Примеры реализации декораторов.

Тема 2.4. Модули стандартной библиотеки языка Python.
1. Понятие модуля.
2. Модули в Python.
3. Описание стандартной библиотеки языка Python.
4. Примеры функций модуля math.
5. Примеры функций модуля random.
6. Примеры функций модуля time.
7. Примеры функций модуля itertools.
8. Получение информации о содержании модуля.

Раздел 3. Работа с коллекциями
Тема 3.1. Множества. Кортежи.
1. Введение в абстрактные структуры данных.
2. Понятие множества и кортежа.
3. Множества и кортежи в языке Python.

4. Синтаксис создания множества.
5. Синтаксис создания кортежа.
6. Работа с элементами множества и кортежа.
7. Групповые операции над множествами.

Тема 3.2. Строки. Списки. Срезы.
1. Понятие о структуре данных «Список».
2. Методы списка.
3. Операции над списком.
4. Понятие среза.
5. Формирование среза из списка.
6. Дополнительные функции и методы работы со строками.
7. Спецификаторы формата, формирование строк.
8. Формирование среза из строки.

Тема 3.3. Методы split и join. Списочные выражения.
1. Назначение, синтаксис и примеры использования метода split.
2. Назначение, синтаксис и примеры использования метода join.
3. Понятие списочного выражения.
4. Синтаксис и примеры использования списочных выражений в Python.

Тема 3.4. Словари. Функции map и filter.
1. Понятие о структуре данных «Словарь».
2. Синтаксис создания словаря в Python.
3. Методы словаря.
4. Операции над словарем.
5. Функция map: назначение, синтаксис, примеры использования.
6. Функция filter: назначение и синтаксис, примеры использования.

Раздел 4. Объектно-ориентированное программирование
Тема 4.1. Введение в технологии программирования. ООП. Полиморфизм.
1. Перечень основных технологий программирования.
2. Характерные особенности структурного программирования.
3. Характерные особенности функционального программирования.
4. Характерные особенности объектно-ориентированного программирования (ООП).
5. Основные понятия ООП: полиморфизм, наследование, инкапсуляция, класс, метод, область видимости, экземпляр класса, конструктор, деструктор.
6. Определение класса в Python.
7. Синтаксис и семантика полиморфизма в Python.

Тема 4.2. ООП. Определение операторов.
1. Синтаксис и семантика определения оператора класса.
2. Наиболее часто перегружаемые операции.

Тема 4.3. ООП. Наследование.
1. Одиночное и множественное наследование классов.
2. Синтаксис и семантика наследования класса в Python.

Тема 4.4. Проектирование и разработка классов
1. Алгоритм проектирования класса.

2. Реализация проекта класса в коде на языке Python.
3. Понятие иерархии классов.
4. Введение в UML.
5. Введение в проектирование иерархии классов.

5.1.3 Перечень лабораторных работ и практических (семинарских) занятий

№ темы	Наименование практических занятий
1.1	Основы формализации и алгоритмизации задач.
1.2	Введение в методологии разработки программного обеспечения. Первое знакомство с языком Python.
1.3	Реализация разветвляющихся алгоритмов в Python. Операции и выражения.
1.4	Реализация итерационных алгоритмов в Python.
2.1	Основы работы с функциями в Python.
2.2	Разработка функций с переменным числом аргументов.
2.3	Использование функции как объекта в Python. Анонимные функции. Декораторы.
2.4	Модули стандартной библиотеки языка Python.
3.1	Основы работы с объектами типа «множество» и «кортеж».
3.2	Основы работы со структурой данных «список». Дополнительные методы работы со строками. Формирование срезов.
3.3	Методы split и join. Списочные выражения.
3.4	Основы работы со структурой данных «словарь». Функции для работы с итерируемыми объектов.
4.1	Введение в технологии программирования. Полиморфизм в объектно-ориентированном программировании.
4.2	Переопределение стандартных операторов.
4.3	Наследование в объектно-ориентированном программировании.
4.4	Проектирование и разработка классов.

5.2 Рабочая программа дисциплины «Работа с данными на языке Python»

5.2.1 Структура дисциплины

№ п/п	Наименование элементов программы	Всего часов	Практические/ лабораторные работы	Сам. раб.	Формируемые компетенции
1.	**Раздел 1. Основы анализа и визуализация данных**				
1.1.	Библиотека NumPy: вычислительные задачи	6	2	4	ПК-2
1.2.	Подготовка и первичный анализ данных с Pandas	10	2	8	ПК-2
1.3.	Чтение данных из различных источников	10	2	8	ПК-2
1.4.	Методы визуализации данных Matplotlib	6	2	4	ПК-2
2.	**Раздел 2. Статистическая обработка данных**				
2.1.	Случайные события и величины. Распределения	10	2	8	ПК-2
2.2.	Описательная статистика	10	2	8	ПК-2
2.3.	Корреляция и корреляционный анализ	10	2	8	ПК-2
2.4.	Проверка гипотез	10	2	8	ПК-2
3.	**Раздел 3. Основы машинного обучения**				
3.1.	Подготовка данных. Нормализация. Категоризация признаков	10	2	8	ПК-2
3.2.	Регрессионный анализ. Линейная и логистическая регрессия	10	2	8	ПК-2
3.3.	Задачи классификации и кластеризации	10	2	8	ПК-2
3.4.	Оценка качества модели машинного обучения	10	2	8	ПК-2
4.	**Раздел 4. Работа с реляционными СУБД**				
4.1.	Основы реляционных СУБД. Отношения. Типы данных	10	2	8	ПК-3
4.2.	Ограничения целостности. Первичный и внешний ключи. Индексация	10	2	8	ПК-3

4.3.	Язык SQL. Соединение таблиц. Группировка. Агрегатные функции	6	2	4	ПК-3
4.4.	Работа с базами данных на Python	6	2	4	ПК-3
	ИТОГО:	**144**	**32**	**112**	

5.2.2 Содержание дисциплины

Раздел 1. Основы анализа и визуализация данных

Тема 1.1. Библиотека NumPy: вычислительные задачи

1. Основы линейной алгебры.
2. Типы данных и массивы NumPy. Основные методы Ndarray.
3. Операции над массивами.
4. Запись и расчёт формул в NumPy.

Тема 1.2. Подготовка и первичный анализ данных с Pandas

1. Структуры данных Pandas: Series и DataFrame.
2. Индексирование и выборка данных.
3. Импорт и экспорт.
4. Сортировка, слияние, объединение, сравнение, решейп датафреймов.
5. Очистка и обработка отсутствующих данных.
6. Первичная визуализация для разведывательного анализа.

Тема 1.3. Чтение данных из различных источников

1. Текстовые и двоичные форматы.
2. Подключение к БД.
3. Импорт и экспорт данных NumPy и Pandas.

Тема 1.4. Методы визуализации данных Matplotlib

1. Типы графиков и их построение с помощью Matplotlib.
2. Надстройка: Seaborn.
3. Сохранение графиков

Раздел 2. Статистическая обработка данных

Тема 2.1. Случайные события и величины. Распределения

1. Случайные события и вероятность.
2. Случайная величина и ее закон распределения.
3. Числовые характеристики случайной величины.
4. Свойства математического ожидания и дисперсии.
5. Нормальный и биномиальный законы распределения.

Тема 2.2. Описательная статистика

1. Графическое представление данных.
2. Табличное представление данных.
3. Расчет обобщающих статистик.

Тема 2.3. Корреляция и корреляционный анализ

1. Параметрические параметры корреляции.
2. Непараметрические параметры корреляции.
3. Построение корреляционного поля и составление корреляционной таблицы.
4. Вычисление выборочных коэффициентов корреляции и корреляционных отношений.

Тема 2.4. Проверка гипотез

1. Классификация статистических гипотез.
2. Проверка параметрических гипотез.
3. Проверка непараметрических гипотез.

Раздел 3. Основы машинного обучения

Тема 3.1. Подготовка данных. Нормализация. Категоризация признаков

1. Виды данных, типы признаков.
2. Очистка данных.
3. Трансформация данных.
4. Интеграция данных.
5. Сокращение данных.

Тема 3.2. Регрессионный анализ. Линейная и логистическая регрессия

1. Понятие регрессии, связь с корреляционными параметрами.
2. Линейная регрессия.
3. Логистическая регрессия.
4. Полиномиальная и степенная регрессия.

Тема 3.3. Задачи классификации и кластеризации

1. Классификация по методу k ближайших соседей.
2. Древовидные модели. Случайный лес. Бэггинг.
3. Кластеризация на основе k средних.
4. Иерархические модели кластеризации.

Тема 3.4. Оценка качества модели машинного обучения

1. Метрики качества регрессионных моделей.
2. Метрики для моделей классификации и кластеризации.

Раздел 4. Работа с реляционными СУБД

Тема 4.1. Основы реляционных СУБД. Отношения. Типы данных

1. Назначение и основные функции СУБД.
2. Модели БД. Сетевая и иерархическая.
3. Реляционная модель данных.
4. Отношения. Свойства отношений.

Тема 4.2. Ограничения целостности. Первичный и внешний ключи. Индексация

1. Понятие ключа отношения: первичный и внешний ключ.
2. Связи между таблицами. Ограничение целостности внешнего ключа. Каскадные операции.
3. Отношения и атрибуты: транзитивные, неполные функциональные и многозначные зависимости атрибутов.
4. Нормализация.

Тема 4.3. Язык SQL. Соединение таблиц. Группировка. Агрегатные функции

1. Запросы к БД. Разновидности запросов и их назначение. Операторы INSERT, SELECT, UPDATE, DELETE.
2. Соединение таблиц. Внешнее и внутреннее условное соединение. Оператор JOIN.
3. Группировка данных. Оператор GROUP BY.
4. Агрегатные функции. Операторы AVG, SUM, MIN, MAX, COUNT

Тема 4.4. Работа базами данных на Python

1. Подключение и получение данных из локальных и серверных БД: Sqlite, MySQL, PostgreSQL.
2. ORM. Модель данных SqlAlchemy.

5.2.3 Перечень лабораторных работ и практических (семинарских) занятий

№ темы	Наименование практических занятий

1.2, 1.3	Первичный анализ данных
1.1, 1.4	Решение расчётных задач. Визуализация результатов
2.1, 2.2	Расчет обобщающих статистик
2.3, 2.4	Построение матриц корреляций. Проверка гипотез
3.1	Разведочный анализ данных
3.2, 3.4	Регрессия
3.3, 3.4	Классификация и кластеризация
4.1, 4.2	Проектирование БД
4.3, 4.4	Построение, заполнение и использование БД. SQL и ORM подходы

5.3 Рабочая программа дисциплины «Разработка веб-приложений на языке Python»

5.3.1 Структура дисциплины

№ п/п	Наименование элементов программы	Всего часов	Практические/ лабораторные работы	Сам. раб.	Формируемые компетенции
1.	**Раздел 1. Основы веб-технологий**				
1.1.	Контроль версий. Репозитории. Git	6	2	4	ПК-1, ПК-3
1.2.	Основы веб-технологий. Протокол HTTP	10	2	8	ПК-1, ПК-3
1.3.	Установка и настройка веб-сервера	10	2	8	ПК-1, ПК-3
1.4.	Выполнение Post и Get-запросов на Python. JSON. Работа с API	6	2	4	ПК-1, ПК-3
2.	**Раздел 2. Фреймворк Flask**				
2.1.	Основы Flask. Установка и настройка.	10	2	8	ПК-1, ПК-3
2.2.	Обработка форм на Flask	10	2	8	ПК-1, ПК-3
2.3.	Архитектура MVC	10	2	8	ПК-1, ПК-3
2.4.	Шаблоны. flask-wtf	10	2	8	ПК-1, ПК-3

3.	**Раздел 3. Разработка веб-приложений на Flask**				
3.1.	ORM-модели на Flask. Sqlalchemy	10	2	8	ПК-1, ПК-3
3.2.	Валидация форм на Flask	10	2	8	ПК-1, ПК-3
3.3.	Аутентификация пользователей	10	2	8	ПК-1, ПК-3
3.4.	Реализация REST API	10	2	8	ПК-1, ПК-3
4.	**Раздел 4. Развертывание и сопровождение веб-приложений**				
4.1.	Хостинг сайтов и приложений. DNS	10	2	8	ПК-1, ПК-3
4.2.	Основы ОС Linux. Работа с удаленным сервером	10	2	8	ПК-1, ПК-3
4.3.	Контейнеризация приложений	6	2	4	ПК-1, ПК-3
4.4.	Тестирование приложений. Библиотеки unittest и pytest	6	2	4	ПК-1, ПК-3
	ИТОГО:	**144**	**32**	**112**	

5.3.2 Содержание дисциплины

Раздел 1. Основы веб-технологий.

Тема 1.1. Контроль версий. Репозитории. Git.

1. Назначение и виды систем контроля версий (VCS). Регистрация на GitHub и установка Git.
2. Понятие репозитория, ветки, коммита. Слияние веток и разрешение конфликтов.
3. Команды Git. Синхронизация проекта с удаленным репозиторием.
4. GitFlow. Командная разработка с использованием GitHub.

Тема 1.2. Основы веб-технологий. Протокол HTTP.

1. Основы сетевого взаимодействия компьютеров.Стек протоколов TCP/IP
2. Назначение и основные характеристики протокола HTTP. Структура HTTP запроса и ответа.
3. Методы HTTP. Аутентификация HTTP. Cookies.
4. Передача файлов с использованием HTTP. Кодирование двоичных данных.

Тема 1.3. Установка и настройка веб-сервера.

1. Основы сетевого взаимодействия компьютеров.Стек протоколов TCP/IP.
2. Настройка локального окружения для разработки в среде Windows.
3. Применение модуля Apache htaccess.
4. Развертывание статических веб-сайтов.

Тема 1.4. Выполнение Post и Get-запросов на Python. JSON. Работа с API

1. Структура Post и Get-запросов.
2. Работа с библиотекой requests.
3. Выполнение запросов к открытым API.
4. Работа с JSON-объектами.

Раздел 2. Фреймворк Flask.

Тема 2.1. Основы Flask. Установка и настройка.

1. Установка Flask.
2. Структура приложения Flask.

3. Передача статического контента
4. Верстка страниц с использованием CSS-фреймворка Bootstrap

Тема 2.2. Обработка форм на Flask.
1. Передача и обработка параметров в адресной строке.
2. Типы элементов форм.
3. Обработка различных элементов форм.
4. Загрузка файлов на сервер

Тема 2.3. Архитектура MVC.
1. Общие сведения об архитектуре MVC.
2. Преимущества применения типовой архитектуры приложения
3. Объектно-ориентированная реализация элементов приложения

Тема 2.4. Шаблоны. flask-wtf.
1. Назначение и преимущество применения шаблонов.
2. Условия, циклы, переменные в шаблонах.
3. Наследование шаблонов.
4. Обработка форм с помощью flask-wtf.

Раздел 3. Разработка веб-приложений на Flask.

Тема 3.1. ORM-модели на Flask. Sqlalchemy.
1. Понятие объектно-реляционного мэппинга (ORM).
2. Создание подключения к БД.
3. Генерация таблиц на основе моделей.
4. Выполнение запросов и получение записей.

Тема 3.2. Валидация форм на Flask.
1. Применение CSRF-токенов.
2. Валидация полей ввода на клиенте.
3. Валидация полей ввода на сервере.
4. Отображение результатов валидации

Тема 3.3. Аутентификация пользователей.
1. Библиотека flask-login
2. Создание таблицы пользователей в БД.
3. Форма аутентификации.
4. Проверка авторизации пользователя

Тема 3.4. Реализация REST API.
1. Монолитная и микросервисная архитектура приложений.
2. Создание сервисов с помощью библиотеки blueprint.
3. Библиотека flask-restful и реализация API.
4. Миграции БД и библиотека alembic

Раздел 4. Развертывание и сопровождение веб-приложений.

Тема 4.1. Хостинг сайтов и приложений. DNS.
1. Виды хостинга. Web-хостинг, хостинг приложений.
2. Развертывание приложений на web-хостинге.
3. Развертывание приложений на хостинге с контейнеризацией.
4. Система DNS и регистрация доменных имен.

Тема 4.2. Основы ОС Linux. Работа с удаленным сервером.
1. Основы работы с ОС Linux. Основные команды, работа с файлами.
2. Виртуальные выделенные серверы (VPS) и облачные серверы.
3. Подключение к удаленному серверу Linux.
4. Установка LAMP-стека на удаленном сервере.

Тема 4.3. Контейнеризация приложений.

1. Docker. Понятие образа и контейнера.
2. Основные команды Docker. Управление контейнерами.
3. Подключение внешних томов в Docker.
4. Reverse proxy. Запуск нескольких контейнеров с web-приложениями на одном сервере.

Тема 4.4. Тестирование приложений. Библиотеки unittest и pytest.

1. Назначение и виды тестирования.
2. Проверка синтаксиса и стиля кода. PEP8. Линтер.
3. Тестирование приложений с использованием pytest.
4. Тестирование приложений в IDE Pycharm.

5.3.3 Перечень лабораторных работ и практических (семинарских) занятий

№ темы	Наименование практических занятий
1.1.	Работа с утилитой Git и сервисом Github
1.2.	Анализ работы протокола HTTP
1.3.	Установка и настройка web-сервера
1.4.	Работа с открытым API по получению json-данных
2.1.	Простое Python-приложение на Flask
2.2.	Обработка форм на Flask
2.3, 2.4	Разработка MVC-приложения с использованием шаблонов flask-wtf
3.1.	Работа с БД MySQL с применением ORM-библиотеки sqlalchemy
3.2.	Обработка и валидация форм на Flask
3.3.	Разработка подсистемы аутентификации пользователей
3.4.	Разработка REST API согласно индивидуальному заданию
4.1.	Регистрация доменного имени и хостинга
4.2.	Работа в командной строке ОС Linux
4.3.	Сборка и запуск docker-контейнера с приложением на Python
4.4.	Разработка тестов для приложения на Python

5.4 Рабочая программа практики

Обязательным элементом обучения является прохождение практики в профильной сфере за пределами СурГУ, в которой обучающийся осваивает ДПП ПП. С целью организации практики Университетом заключаются договоры о практической подготовке

5.4.1. Содержание практики

Структура производственной практики предусматривает несколько этапов:
1. Подготовительный этап включает в себя решение всех организационных вопросов: ознакомление обучающихся с целями и задачами производственной практики, программой, отчетной документацией; знакомство с руководителями практики; распределение обучающихся на предприятия для прохождения практики.
2. Основной этап - разработка web-приложения в формате индивидуального или группового проекта.
3. Заключительный этап – подготовка доклада и презентации проекта и участие в заключительной конференции.
На всех этапах практики возрастает роль самостоятельной работы по выполнению заданий программы практики.

6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

6.1 Учебно-методическое и информационное обеспечение

6.2. Материально-техническое оснащение

№ п/п	Наименование объекта, подтверждающего наличие материально-технического обеспечения, с перечнем основного оборудования
1.	Компьютерный класс для проведения лабораторных и практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации оснащена: комплект специализированной учебной мебели, маркерная (меловая) доска; Количество посадочных мест – 25. Технические средства обучения для представления учебной информации: стационарный экран, переносной проектор, 13 компьютеров. Используемое программное обеспечение: Microsoft Windows, пакет прикладных программ Microsoft Office, JetBrains PyCharm Community. Обеспечен доступ к сети Интернет и в электронную информационную среду организации.

6.3. Кадровые условия реализации программы:

Программа переподготовки реализуется штатными преподавателями, имеющими высшее профильное образование в ИТ-отрасли и педагогический стаж работы в университете не менее 3 лет. Не менее 20% от общего объема аудиторных занятий по программе переподготовки реализуется лицами, имеющими стаж в профессии в ИТ-отрасли не менее двух лет.
Итоговая аттестация проводится с обязательным участием представителей профильных организаций-работодателей.

7. ФОРМЫ АТТЕСТАЦИИ

7.1 Промежуточный контроль

В целях определения уровня сформированности/развития цифровых компетенций обучающихся для лиц, заявившихся на обучение, обязательным является прохождение промежуточной аттестации, включающее в себя зачеты по итогам изучения дисциплины и прохождения практики.
Завершение всех этапов промежуточной аттестации является допуском к итоговой аттестации.

7.3 Итоговая аттестация

В ходе итоговой аттестации обучающиеся обеспечивают презентацию (защиту) разработанного цифрового решения (проекта), а также перечня решаемых им проблем и эффектов, ожидаемых от его реализации (внедрения) в отрасль. Проектное решение должно отвечать критериям актуальности, законченности, а также возможности интеграции его компонентов в иные системы и сервисы. Лицам, успешно освоившим ДПП ПП присваивается дополнительная квалификация.
Диплом о профессиональной переподготовке выдается не ранее завершения обучения по основной образовательной программе высшего образования и получения соответствующего документа об образовании и о квалификации (за исключением лиц, имеющих среднее профессиональное или высшее образование)

8. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

8.1 Промежуточный контроль

8.1.1. Задания для промежуточной аттестации по дисциплине Программирование на языке Python

Примерный перечень вопросов для контроля освоения теоретического материала дисциплины:

1. Основные этапы компьютерного решения задачи.
2. Языки формальной спецификации.
3. Функциональные диаграммы, диаграммы переходов состояний.
4. Алгоритм: определение понятия, свойства алгоритма.
5. Способы записи алгоритмов: перечень и ключевые особенности каждого из способов.
6. Блок-схема: основные элементы и правила построения.
7. Разновидности алгоритмов.
8. Методики разработки алгоритмов.
9. Особенности установки и работы в стандартной среде Python.
10. Особенности установки и работы в IDE JetBrains PyCharm.
11. Жизненный цикл программного продукта.
12. Особенности основных методологий разработки программного обеспечения: каскадная модель, V-модель, инкрементная модель, модель быстрой разработки программного обеспечения (RAD-модель), гибкая (Agile) методология, итеративная модель, спиральная модель.
13. Типы данных и переменные в Python.
14. Функции ввода и вывода в Python.
15. Порядок действий для создания графического представления алгоритма на примере онлайн-сервиса draw.io.
16. Арифметические операции и выражения в Python.
17. Способы формирования строк в языке Python.
18. Возможности языка Python по работе строками: функции и методы работы со строками, спецификаторы формата, формирование среза, метод split.
19. Логические операции и выражения в Python.
20. Возможности языка Python по реализации разветвляющихся алгоритмов.
21. Возможности языка Python по реализации циклических алгоритмов.
22. Понятие и назначение функции в программировании.
23. Встроенные функции в Python: примеры (с параметрами и без, с фиксированным и переменным числом аргументов), синтаксис, особенности вызова.
24. Определение пользовательской функции с фиксированным и/или переменным числом позиционных аргументов (привести пример).

25. Определение пользовательской функции с фиксированным и/или переменным числом именованных аргументов (привести пример).
26. Порядок передачи аргументов в функцию.
27. Реализация возврата функцией нескольких значений.
28. Область видимости: определение понятия, особенности работы.
29. Организация доступа к значениям аргументов в функции с переменным числом позиционных и/или именованных аргументов.
30. Особенности работы с функциями как с объектами в языке Python.
31. Лямбда-функция: определение, назначение, синтаксис, примеры использования.
32. Декораторы: назначение, синтаксис, примеры реализации.
33. Модули в Python: примеры стандартных модулей с указанием назначения, синтаксис подключения, синтаксис импорта функции из модуля, способы получения информации о содержании модуля.
34. Описание стандартной библиотеки языка Python.
35. Абстрактные структуры данных в Python: перечень, основные методы, примеры использования.
36. Множества и кортежи в языке Python: синтаксис создания, примеры работы с элементами, групповые операции над множествами.
37. Список в Python: методы, операции над списком, формирование среза.
38. Списочные выражения в Python: синтаксис и примеры использования.
39. Словарь в Python: синтаксис создания, методы, операции над словарем.
40. Функция map: назначение, синтаксис, примеры использования.
41. Функция filter: назначение и синтаксис, примеры использования.
42. Характерные особенности структурного программирования.
43. Характерные особенности функционального программирования.
44. Характерные особенности объектно-ориентированного программирования (ООП).
45. Основные понятия ООП: полиморфизм, наследование, инкапсуляция, класс, метод, область видимости, экземпляр класса, конструктор, деструктор.
46. Определение класса в Python.
47. Синтаксис и семантика полиморфизма в Python.
48. Синтаксис и семантика определения оператора класса.
49. Наследование класса в Python: синтаксис и семантика, одиночное и множественное наследование.
50. Алгоритм проектирования класса.
51. Особенности реализации проекта класса в коде на языке Python.
52. Проектирование иерархии классов.
53. Язык UML: назначение, примеры конструкций языка, используемых при проектировании иерархии классов.

Примерные практические задания для контроля освоения практической части дисциплины.

Общая формулировка задания: В соответствии с представленным описанием задачи необходимо выполнить следующие действия:

– Формализовать поставленную задачу с применением любого из известных методов формализации задач;

– Разработать алгоритм решения поставленной задачи с применением любого из известных методов проектирования алгоритмов. Алгоритм решения задачи следует изобразить графически, используя любые доступные программные средства, обладающие соответствующими возможностями;

– В среде программирования разработать программу с консольным интерфейсом пользователя. Перечень необходимых пунктов интерфейса определить самостоятельно исходя из условия задачи.

Интерфейс должен быть интуитивно понятным в обращении. Ввод исходных данных и вывод результата их обработки должен осуществляться через выбор соответствующих команд консольного интерфейса пользователя, а сама обработка – посредством пользовательской функции (или набора взаимосвязанных функций), определение которой должно содержать комментарии, поясняющие шаги выполняемого алгоритма обработки.

Примеры описания задач промежуточной аттестации:

1. По выбору пункта меню «*Сформировать последовательность*» в консольном приложении формируется последовательность из N случайных чисел. Значение N предварительно указывается пользователем с помощью соответствующей команды в меню программы. После формирования полученная последовательность выводится в окне программы. В меню программы также должны быть предусмотрены пункты, при выборе которых полученная последовательность упорядочивается либо в прямом, либо в обратном порядке и выводится в окне программы под исходной последовательностью.
2. По выбору пункта меню «*Сформировать последовательность*» консольного приложения генерируется последовательность из N случайных чисел, находящихся в интервале от A до B. Значения N, A, B предварительно указываются пользователем с помощью соответствующих команд меню приложения. После генерации последовательность случайных чисел выводится в окне программы. По выбору в программе пункта меню «Обработка» осуществляется вычисление следующих параметров: сумма всех значений последовательности, среднее арифметическое всех элементов последовательности, количество четных и нечетных элементов соответственно, количество положительных и отрицательных значений, количество значений, состоящих из 1, 2, 3 и т.д. разрядов. Каждый из перечисленных параметров выводится в отдельной строке окна программы.
3. После выбора пункта меню «*Ввести последовательность*» консольного приложения пользователь осуществляет ввод последовательности, состоящей из произвольного количества различных типов символов. Количество вводимых элементов заранее не известно. Последовательность при этом может содержать абсолютно произвольные данные, то есть цифры, буквы, знаки математических операций, скобки и т.п. После выбора в консольном приложении пункта меню «*Обработать*», в окне программы осуществляется построчный вывод количества различных типов символов, введенных пользователем, то есть букв, целых чисел, вещественных чисел, знаков математических операций и т.д. Для тех категорий символов, которые не были использованы при вводе, никаких значений не выводится.
4. После выбора пункта меню «*Ввести текст*» консольного приложения пользователь осуществляет ввод нескольких предложений. Количество предложений заранее не известно. После выбора пункта меню «*Вывести статистику текста*» в окне программы должна быть выведена построчно следующая информация о введенном тексте: количество слов в тексте, количество знаков препинания, количество пробелов, количество цифр, количество прочих символов, самое длинное слово в тексте, средняя длина слова в тексте.
5. После выбора пункта меню «*Ввести текст*» консольного приложения пользователь осуществляет ввод нескольких предложений. Количество предложений заранее не известно. После окончания ввода программа после выбора пункта меню «*Развернуть текст*» консольного приложения должна изменить порядок следования предложений на обратный и вывести полученный результат в окне программы.
6. Пользователь через вызов пункта меню «*Задать размерность матрицы*» консольного приложения указывает количество строк и столбцов матрицы. Далее с помощью пункта меню «*Ввести матрицу*» осуществляет построчный ввод её значений. Матрица может иметь произвольные размеры, но должна содержать не менее трех строк и трех столбцов. В меню приложения также должны быть предусмотрены команды «*Сдвинуть значения матрицы по строкам*» и «*Сдвинуть значения матрицы по столбцам*». После вызова какой-либо из данных команд вводятся количества строк или столбцов, на которые необходимо циклически сдвинуть соответствующие элементы

матрицы. Полученная матрица должна быть выведена в окне программы под исходной матрицей после выполнения любого из указанных сдвигов значений.

7. Пользователь через вызов пунктов меню «*Указать количество элементов первого множества*» и «*Указать количество элементов второго множества*» консольного приложения задает значения количества элементов двух числовых множеств. После выбора пункта меню «*Генерация*» консольного приложения обе последовательности заполняются указанными количествами случайных целочисленных значений и выводятся в окне программы построчно. При этом строки, содержащие значения множеств, при выводе в окне приложения должны быть разделены между собой с помощью пустой строки. Кроме пункта меню «*Генерация*» в окне программы также должен присутствовать пункт меню «*Вычисление*». После вызова данного пункта должно появиться дополнительное меню с набором следующих возможных вариантов операций с введенными множествами: пересечение множеств, объединение множеств, разность первого и второго множества, разность второго и первого множества, декартово произведение множеств. После выбора какой-либо из операций с множествами должен быть выведен результат её выполнения. В дополнительном меню с выбором операций над множествами должна быть предусмотрена команда возврата в основное меню.

8.1.2. Задания для промежуточной аттестации по дисциплине Работа с данными на языке Python

1. Типы данных и массивы NumPy. Основные методы Ndarray.
2. Структуры данных Pandas: Series и DataFrame.
3. Сортировка, слияние, объединение, сравнение, решейп датафреймов.
4. Очистка и обработка отсутствующих данных.
5. Первичная визуализация для разведывательного анализа.
6. Случайные события и вероятность.
7. Случайная величина и ее закон распределения.
8. Числовые характеристики случайной величины.
9. Свойства математического ожидания и дисперсии.
10. Нормальный и биномиальный законы распределения.
11. Графическое представление данных.
12. Табличное представление данных.
13. Расчет обобщающих статистик.
14. Параметрические параметры корреляции.
15. Непараметрические параметры корреляции.
16. Классификация статистических гипотез.
17. Виды данных, типы признаков.
18. Очистка данных.
19. Трансформация данных.
20. Интеграция данных.
21. Сокращение данных.
22. Понятие регрессии, связь с корреляционными параметрами.
23. Линейная регрессия.
24. Классификация по методу k ближайших соседей.
25. Древовидные модели. Случайный лес. Бэггинг.
26. Кластеризация на основе k средних.
27. Иерархические модели кластеризации.
28. Метрики качества регрессионных моделей.
29. Метрики для моделей классификации и кластеризации.
30. Назначение и основные функции СУБД.
31. Модели БД. Сетевая и иерархическая.

32. Реляционная модель данных.
33. Отношения. Свойства отношений.
34. Понятие ключа отношения: первичный и внешний ключ.
35. Связи между таблицами. Ограничение целостности внешнего ключа. Каскадные операции.
36. Отношения и атрибуты: транзитивные, неполные функциональные и многозначные зависимости атрибутов.
37. Запросы к БД. Разновидности запросов и их назначение. Операторы INSERT, SELECT, UPDATE, DELETE.
38. Соединение таблиц. Внешнее и внутреннее условное соединение. Оператор JOIN.
39. Группировка данных. Оператор GROUP BY.
40. Агрегатные функции. Операторы AVG, SUM, MIN, MAX, COUNT

8.1.3. Задания для промежуточной аттестации по дисциплине Разработка веб-приложений на Python

1. Напишите функцию, которая формирует HTML-страницу с текстом ‘Hello, world!’ при доступе к корневой папке сайта.
2. Напишите код, обрабатывающий POST запрос и выводящий на страницу все переданные параметры и их значения.
3. Напишите код, обрабатывающий GET запрос и выводящий на страницу все переданные параметры и их значения.
4. Имеется страница с формой. Какой код сможет получить значение, введенное в поле username?

```
<form action="/login" method="post">
    <p><input type=text name=username>
    <p><input type=submit value=Login>
</form>
```

5. Имеется код. Какие URL смогут вызвать функцию show_post и выполниться без ошибок?

```
@app.route('/post/<int:post_id>')
def show_post(post_id):
    return 'Post %d' % post_id
```

6. По какому URL будет доступно веб-приложение, запущенное следующим кодом:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
app.run()
```

7. В какую папку необходимо положить файл logo.png чтобы он был доступен через URL /logo:

```
@app.route('/logo')
def image():
    return f'''<img src="{url_for('static',
filename='img/logo.png')}">'''
```

8. Напишите Flask-функцию для генерации страницы с формой для входа в систему с полями “логин”, “пароль” и кнопкой с текстом “Войти”, а также функцию-обработчик, получающую значения полей через POST-запрос и проверяющую правильность логина и пароля (правильный логин и пароль задаются константами)
9. При помощи Flask-функции реализуйте HTML-форму для загрузки файла на сервер, а также функцию-обработчик, сохраняющую загруженный файл в определенной папке на сервере.
10. Напишите функцию, которая формирует HTML-страницу с текстом ‘Hello, world!’ при доступе к корневой папке сайта с использованием Flask-шаблонизатора

11. Напишите код, обрабатывающий POST запрос и выводящий на страницу все переданные параметры и их значения. Значения параметров должны передаваться в шаблон.
12. Напишите код, обрабатывающий GET запрос и выводящий на страницу все переданные параметры и их значения. Значения параметров должны передаваться в шаблон.
13. Напишите Flask-функцию для генерации страницы с формой для входа в систему с полями “логин”, “пароль” и кнопкой с текстом “Войти”, а также функцию-обработчик, получающую значения полей через POST-запрос и проверяющую правильность логина и пароля (правильный логин и пароль задаются константами). Результат аутентификации должен передаваться в шаблон как параметр и проверяться в шаблоне при помощи условия.

8.2 Итоговая аттестация

Итоговая аттестация проводится на основе принципов объективности и независимости оценки качества подготовки обучающихся. К итоговой аттестации допускается обучающийся, не имеющий задолженности и в полном объеме выполнивший учебный план по программе. Итоговая аттестация является обязательной для обучающихся завершающих обучение. Оценка качества освоения программы проводится в отношении соответствия результатов освоения программы, заявленным целям и планируемым результатам обучения.
Обучающиеся успешно прошедшие итоговую аттестацию получают диплом о профессиональной переподготовке (при наличии основного профессионального образования). В случае одновременного получения образования по основной профессиональной программе и дополнительной программе переподготовки диплом о профессиональной переподготовке выдается одновременно с дипломом о профессиональном образовании. Обучающиеся, не прошедшие итоговую аттестацию по уважительной причине (подтвержденной документально) предоставляется возможность пройти итоговую аттестацию в сроки, определяемые Университетом. Обучающиеся не прошедшие итоговую аттестацию или получившие на итоговой аттестации неудовлетворительные результаты, выдается справка об обучении или периоде обучения.
Итоговая аттестация может проводится с использованием электронного обучения, дистанционных образовательных технологий. При проведении итоговых аттестационных испытаний с применением электронного обучения, дистанционных образовательных технологий Университет обеспечивает идентификацию личности обучающихся.
Итоговая аттестация осуществляется аттестационной комиссией. Состав аттестационной комиссии утверждается распоряжением проректора по учебно-методической работе. Основные функции аттестационной комиссии:
– комплексная оценка уровня знаний и умений, компетенций обучающихся установленных программой;
– рассмотрение вопросов о предоставлении обучающимся по результатам освоения образовательной программы права заниматься профессиональной деятельностью.

Итоговая аттестация осуществляется в виде защиты итоговой аттестационной работы, представляющей конкретный результат деятельности обучающегося в ходе освоения образовательной программы и имеющий прикладную направленность.

Обучающимся предоставляется право выбора темы итоговой аттестационной работы или обучающийся может предложить свою тему с обоснованием целесообразности ее разработки.
Защита итоговой аттестационной работы проходит в виде доклада. Результаты защиты итоговой аттестационной работы объявляются в день защиты и оформляются протоколами.
По результатам защиты итоговой аттестационной работы выставляются отметки по четырехбальной шкале: “отлично”, “хорошо”, “удовлетворительно”, “неудовлетворительно”.
Отметка “неудовлетворительно” выставляется обучающемуся, показавшему пробелы в знании основного содержания, предусмотренного образовательной программой, допустившего серьезные ошибки в выполнении предусмотренных программой заданий.

Отметка “удовлетворительно” выставляется обучающемуся показавшему знание основного учебно-программного материала в объеме, необходимомо для дальнейшего обучения и профессиональной деятельности, но допустившего погрешности в ответах на аттестационном испытании.
Отметки “хорошо” заслуживает обучающийся показавший полное знание программного материала, способный к самостоятельному пополнению знаний в ходе дальнейшего обучения и профессиональной деятельности.
Отметки “отлично” заслуживает обучающийся показавший всестороннее и глубокое знание программного материала, умение решать поставленные задачи, проявивший творческие способности в понимании и применении учебно-программного материала.
Примерные темы итоговой аттестационной работы:

1. Web-приложение для подбора вакансий и резюме с использованием технологий машинного обучения
2. Онлайн-платформа для размещения предложений по оказанию услуг и поиска исполнителей
3. Информационная система для организации научных конференций
4. Приложение для проведения собраний собственников многоквартирного дома
5. Web-сервис для валидации аватаров пользователей
6. Web-приложение для подбора кулинарных рецептов по заданным параметрам
7. Приложение для учета совместных расходов, управляемое с помощью мессенджера Telegram
8. Автоматизированная информационная система управления прокатом оборудования
9. Web-приложение для организации рынка долгосрочной аренды жилья
10. Web-приложение для учета посещения занятий и мероприятий с использованием мобильных устройств
11. Сервис мгновенного обмена сообщениями, встраиваемый в произвольный web-сайт
12. Web-сервис для проведения мгновенных опросов и голосований

9. ОЦЕНКА КАЧЕСТВА РЕАЛИЗАЦИИ ПРОГРАММЫ

Внутренняя оценка качества реализации включает в себя результаты текущего, промежуточного контроля по освоению дисциплин учебного плана, результаты итогового контроля по освоению программы, анкетирование слушателей для оценки удовлетворенности качеством реализации дополнительной образовательной программы.

Эффективность реализации ДПП ПП определяется, в том числе динамикой прохождения процедуры комплексной и итоговой оценки (ассесмента) обучающимися:

1. Высокая степень эффективности реализации ДПП ПП подтверждается положительной (отличной от нуля) динамикой по итогам промежуточной комплексной оценки (ассесмента) и достижением заявленного в ДПП ПП уровня развития компетенций по результатам итоговой оценки (ассесмента) не менее чем у 80% обучающихся.
2. Средняя степень эффективности реализации ДПП ПП подтверждается положительной (отличной от нуля) динамикой по итогам промежуточной комплексной оценки (ассесмента) и достижением заявленного в ДПП ПП уровня развития компетенций по результатам итоговой оценки (ассесмента) не менее чем у 50% обучающихся.

Низкая степень эффективности реализации ДПП ПП определяется положительной (отличной от нуля) динамикой по итогам промежуточной комплексной оценки (ассесмента) и достижением заявленного в ДПП ПП уровня развития компетенций по результатам итоговой оценки (ассесмента) менее чем у 50% обучающихся.